【Tornado】API接口使用Basic Auth认证

请叫我算术嘉 2020-05-03 22:33:13 4253 收藏

分类专栏： Tornado python 文章标签： tornado python basic auth

版权

Tornado

同时被 2 个专栏收录

2 篇文章0 订阅

订阅专栏

python

118 篇文章0 订阅

订阅专栏

1、拿到认证请求

2、解码

3、与数据库中的用户进行比对

4、如果请求没有携带basic auth信息，浏览器弹框输入

5、basic信息错误，还是继续弹框输入

class BasicAuthHandler(tornado.web.RequestHandler):

def initialize(self, db):

self.db = db

def create\_auth\_header(self):

self.set\_status(401)

self.set\_header('WWW-Authenticate', 'Basic realm=Restricted')

self.\_transforms = []

self.finish()

def get(self):

db = self.db

cursor = db.cursor(pymysql.cursors.DictCursor)

# Authorization: Basic base64("user:passwd")

auth\_header = self.request.headers.get('Authorization', None)

if auth\_header is not None:

# Basic Zm9vOmJhcg==

auth\_mode, auth\_base64 = auth\_header.split(' ', 1)

assert auth\_mode == 'Basic'

# Zm9vOmJhcg解码

auth\_info = base64.b64decode(auth\_base64)

# byte转str

auth\_username, auth\_password = auth\_info.decode('utf-8').split(":")

try:

name = auth\_username

cursor.execute(

"SELECT \* FROM blog\_bloguser WHERE name='{}'".format(name)

)

result = cursor.fetchone()

if result is not None:

password = result['password']

if auth\_password == password:

self.create\_auth\_header()

else:

self.create\_auth\_header()

else:

self.create\_auth\_header()

except Exception as e:

return self.write(e)

else:

self.create\_auth\_header()

发现一个库，可以很好地封装

https://pypi.org/project/tornado-basic-auth/

只要在接口上加上装饰器，装饰器的入参是一个函数，该函数接收的参数就是basic auth的用户名，密码，拿到用户名后去数据库里查一下，函数返回bool类型，True或者False

def basic\_auth\_valid(user, pwd):

cursor = mysqldb.cursor(pymysql.cursors.DictCursor)

try:

cursor.execute(

"SELECT \* FROM blog\_bloguser WHERE name='{}'".format(user)

)

result = cursor.fetchone()

if result is not None:

password = result['password']

if pwd == password:

return True

else:

return False

else:

return False

except Exception as e:

return False

@basic\_auth(basic\_auth\_valid)

class GetALlBlog(tornado.web.RequestHandler):

def initialize(self, db):

self.db = db

print("db is ok")

def get(self):

db = self.db

cursor = db.cursor(pymysql.cursors.DictCursor)

try:

cursor.execute(

"SELECT A.id, A.title, A.`timestamp`, A.views, A.greats, A.comments,U.name as 'authorname' FROM blog\_articles A, blog\_bloguser U WHERE A.authorname\_id = U.id AND A.STATUS = '有效'LIMIT 10"

)

result = cursor.fetchall()

return\_data = {}

return\_data["code"] = 200

return\_data["message"] = "success"

return\_data["data"] = result

self.finish(json.dumps(return\_data, cls=DateEncoder))

except Exception as e:

return self.write(e)

db.commit()

cursor.close()

————————————————

版权声明：本文为CSDN博主「请叫我算术嘉」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/ssjdoudou/article/details/105909476